A Conceptional Lego Toolbox for Bayesian Distributional Regression Models

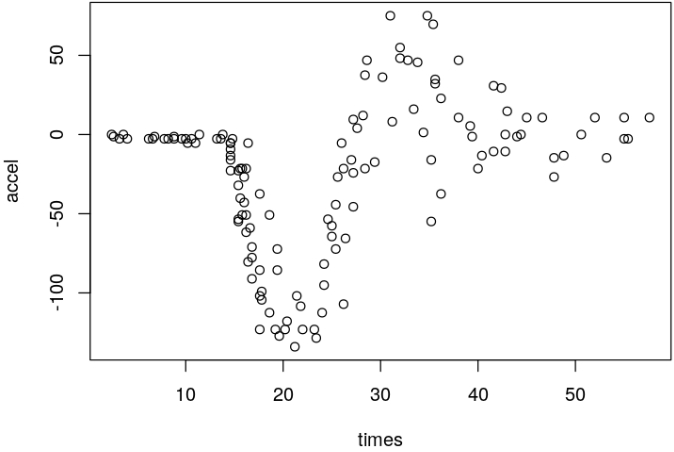
Nikolaus Umlauf, Nadja Klein, Meike Köhler, Sonja Greven, Achim Zeileis

http://eeecon.uibk.ac.at/~umlauf/

Introduction

Example: Head acceleration in a simulated motorcycle accident

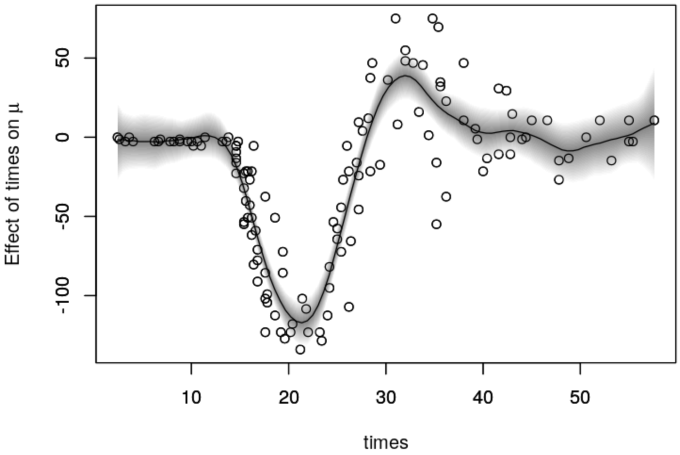
$$\texttt{accel} \sim N(\mu, \sigma^2).$$



Introduction

Example: Head acceleration in a simulated motorcycle accident

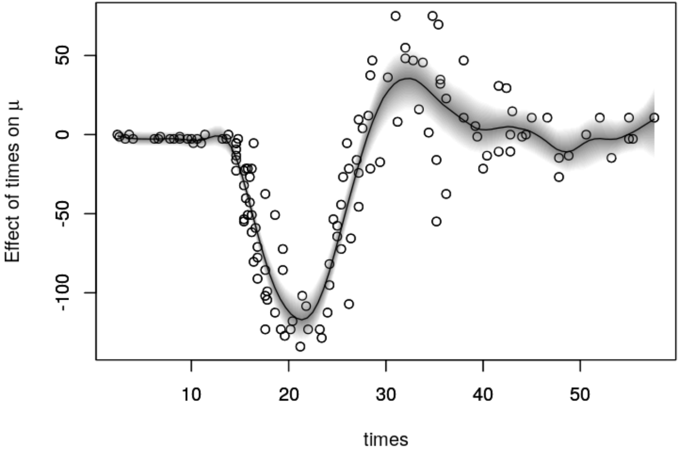
$$\texttt{accel} \sim N(\mu = f(\texttt{times}),\ log(\sigma^2) = \beta_0).$$



Introduction

Example: Head acceleration in a simulated motorcycle accident

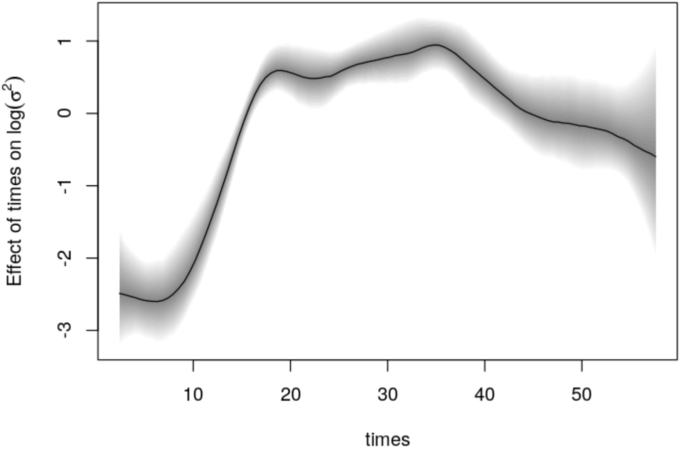
$$\texttt{accel} \sim N(\mu = f(\texttt{times}),\, log(\sigma^2) = f(\texttt{times})).$$



Introduction

Example: Head acceleration in a simulated motorcycle accident

$$\texttt{accel} \sim N(\mu = f(\texttt{times}),\, log(\sigma^2) = f(\texttt{times})).$$



Distributional regression

Model structure

Any parameter of a population distribution $\mathcal{D}$ may be modeled by explanatory variables

$$y \sim \mathcal{D}\left(h_1(\theta_1) = \eta_1,\ h_2(\theta_2) = \eta_2, \ldots,\ h_K(\theta_K) = \eta_K\right),$$

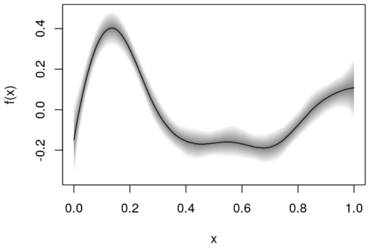
Each parameter is linked to a structured additive predictor

$$h_k(\theta_k) = \eta_k = \eta_k(\mathbf{x}; \boldsymbol{\beta}_k) = f_{1k}(\mathbf{x}; \boldsymbol{\beta}_{1k}) + \ldots + f_{J_k k}(\mathbf{x}; \boldsymbol{\beta}_{J_k k}),$$

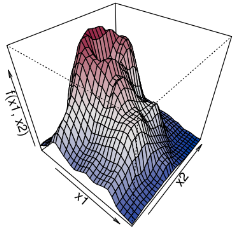
$j = 1, \ldots, J_k$ and $k = 1, \ldots, K$ and $h_k(\cdot)$ are known monotonic link functions.

Distributional regression
Functional types

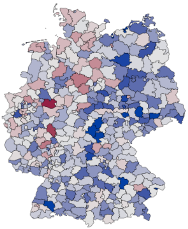
Nonlinear effects of continuous covariates



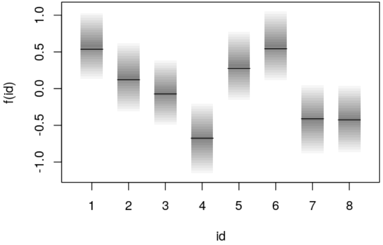
Two-dimensional surfaces



Spatially correlated effects f(x) = f(s)



Random intercepts f(x) = f(id)



Bayesian regression

A **not** complete list of software packages dealing with Bayesian regression models:

- **bayesm**, univariate and multivariate, SUR, multinomial logit, ...
- **bayesSurv**, survival regression, ...
- **MCMCpack**, linear regression, logit, ordinal probit, probit, Poisson regression, ...
- **MCMCglmm**, generalized linear mixed models (GLMM).
- **spikeSlabGAM**, Bayesian variable selection, model choice, in generalized additive mixed models (GAMM), ...
- **gammSlice**, generalized additive mixed models (GAMM).
- **BayesX**, structured additive distributional regression (STAR), ...
- **INLA**, generalized additive mixed models (GAMM), ...
- **WinBUGS**, **JAGS**, **STAN**, general purpose sampling engines.

$\vdots$

Basic ideas

- Design framework to fit models with **different estimation engines** (Bayesian or frequentist).
- **Integration** of new and existing code, as well as interfacing, as easy as possible.
- Symbolic descriptions that do **not** restrict to any specific type of model and term structure.
- Specialized/optimized engines to apply Bayesian **structured additive distributional regression** a.k.a. Bayesian additive models for location scale and shape (**BAMLSS**) and beyond.
- **Maximum flexibility/extendability**, also concerning functional types.

A conceptional Lego toolbox

Common priors

For simple linear effects $\mathbf{X}_{jk}\boldsymbol{\beta}_{jk}$: $p(\boldsymbol{\beta}_{jk}) \propto \mathit{const}$.

For the smooth terms:

$$p(\boldsymbol{\beta}_{jk}) \propto \left(\frac{1}{\tau_{jk}^2}\right)^{rk(\mathbf{K}_{jk})/2} \exp\left(-\frac{1}{2\tau_{jk}^2}\boldsymbol{\beta}_{jk}^\top \mathbf{K}_{jk}\boldsymbol{\beta}_{jk}\right),$$

where $\mathbf{K}_{jk}$ is a quadratic penalty matrix.

Priors $p(\tau_{jk}^2)$: IG, half-Cauchy, half-normal, uniform priors, etc.

A conceptional Lego toolbox
Model fitting

The main building block of regression model algorithms is the probability density function $f(\mathbf{y}|\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K)$.

Estimation typically requires to evaluate

$$\begin{aligned}\ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) &= \sum_{i=1}^{n} \log f(y_i; \theta_{i1} = h_1^{-1}(\eta_{i1}(\mathbf{x}_i, \boldsymbol{\beta}_1)), \dots \\ &\qquad \dots, \theta_{iK} = h_K^{-1}(\eta_{iK}(\mathbf{x}_i, \boldsymbol{\beta}_K))),\end{aligned}$$

with $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \dots, \boldsymbol{\beta}_K^\top)^\top$ and $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_K)$.

The log-posterior

$$\log p(\boldsymbol{\vartheta}; \mathbf{y}, \mathbf{X}) \propto \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) + \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left\{ \log p_{jk}(\boldsymbol{\vartheta}_{jk}) \right\},$$

where, e.g., $\boldsymbol{\vartheta}_{jk} = (\boldsymbol{\beta}_{jk}^\top, (\boldsymbol{\tau}_{jk}^2)^\top)^\top$
(frequentist, penalized log-likelihood).

A conceptional Lego toolbox
Model fitting

Posterior mode estimation, fortunately, partitioned updating is possible

$$\begin{aligned}
\boldsymbol{\beta}_1^{(t+1)} &= U_1(\boldsymbol{\beta}_1^{(t)}, \boldsymbol{\beta}_2^{(t)}, \ldots, \boldsymbol{\beta}_K^{(t)}) \\
\boldsymbol{\beta}_2^{(t+1)} &= U_2(\boldsymbol{\beta}_1^{(t+1)}, \boldsymbol{\beta}_2^{(t)}, \ldots, \boldsymbol{\beta}_K^{(t)}) \\
&\vdots \\
\boldsymbol{\beta}_K^{(t+1)} &= U_K(\boldsymbol{\beta}_1^{(t+1)}, \boldsymbol{\beta}_2^{(t+1)}, \ldots, \boldsymbol{\beta}_K^{(t)}),
\end{aligned}$$

E.g., Newton-Raphson type updating

$$\boldsymbol{\beta}_k^{(t+1)} = U_k(\boldsymbol{\beta}_k^{(t)} | \cdot) = \boldsymbol{\beta}_k^{(t)} - \mathbf{H}_{kk}\left(\boldsymbol{\beta}_k^{(t)}\right)^{-1} \mathbf{s}\left(\boldsymbol{\beta}_k^{(t)}\right).$$

Can be further partitioned for each function within parameter block k. Moreover, using a basis function approach yields IWLS updates

$$\boldsymbol{\beta}_{jk}^{(t+1)} = (\mathbf{X}_{jk}^\top \mathbf{W}_{kk} \mathbf{X}_{jk} + \tau_{jk}^{-2} \mathbf{K}_{jk})^{-1} \mathbf{X}_{jk}^\top \mathbf{W}_{kk} (\mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t)}).$$

A conceptional Lego toolbox

Model fitting

MCMC simulation

- Random walk Metropolis, symmetric $q(\boldsymbol{\beta}_{jk}^{\star}|\boldsymbol{\beta}_{jk}^{(t)})$.
- Derivative based MCMC, second order Taylor series expansion centered at the last state $p(\boldsymbol{\beta}_{jk}^{\star}|\cdot)$ yields $N(\boldsymbol{\mu}_{jk}^{(t)}, \boldsymbol{\Sigma}_{jk}^{(t)})$ proposal with

$$\begin{aligned}\left(\boldsymbol{\Sigma}_{jk}^{(t)}\right)^{-1} &= -\mathbf{H}_{kk}\left(\boldsymbol{\beta}_{jk}^{(t)}\right)\\ \boldsymbol{\mu}_{jk}^{(t)} &= \boldsymbol{\beta}_{jk}^{(t)} - \mathbf{H}_{kk}\left(\boldsymbol{\beta}_{jk}^{(t)}\right)^{-1}\mathbf{s}\left(\boldsymbol{\beta}_{jk}^{(t)}\right).\end{aligned}$$

  Metropolis-Hastings acceptance probability

$$\alpha\left(\boldsymbol{\beta}_{jk}^{\star}|\boldsymbol{\beta}_{jk}^{(t)}\right) = \min\left\{\frac{p(\boldsymbol{\beta}_{jk}^{\star}|\cdot)q(\boldsymbol{\beta}_{jk}^{(t)}|\boldsymbol{\beta}_{jk}^{\star})}{p(\boldsymbol{\beta}_{jk}^{(t)}|\cdot)q(\boldsymbol{\beta}_{jk}^{\star}|\boldsymbol{\beta}_{jk}^{(t)})}, 1\right\}.$$

- Other sampling schemes, e.g., slice sampling, NUTS, t-walk, ... ?!

A conceptional Lego toolbox

Model fitting

The following “lego bricks” are repeatedly used within BAMLSS candidate algorithms:

For log-posterior (likelihood):

- Density function of response distribution $\mathcal{D}$,
- link functions $h_k(\cdot)$,
- priors.

For e.g., IWLS fitting or derivative based MCMC:

- First and second order derivatives of log-posterior,
- often can be put together from derivatives of log-density, link functions and log-priors.

A conceptional Lego toolbox

Algorithm

A simple generic algorithm for BAMLSS models:

```
while(eps > ε & t < maxit) {
    for(k in 1:K) {
        for(j in 1:J[k]) {
            Compute η̃ = η_k − f_jk.
            Obtain new (β*_jk, (τ²_jk)*)ᵀ = U_jk(X_jk, y, η̃, β_jk^[t], (τ²_jk)^[t]).
            Update η_k.
        }
    }
    t = t + 1
    Compute new eps.
}
```

That is, in each inner step: compute $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_{\mathtt{k}} - \mathbf{f}_{\mathtt{jk}}$; obtain new $(\boldsymbol{\beta}^{\star}_{\mathtt{jk}}, (\tau^2_{\mathtt{jk}})^{\star})^{\top} = \mathtt{U}_{\mathtt{jk}}(\mathbf{X}_{\mathtt{jk}}, \mathbf{y}, \tilde{\boldsymbol{\eta}}, \boldsymbol{\beta}^{[\mathtt{t}]}_{\mathtt{jk}}, (\tau^2_{\mathtt{jk}})^{[\mathtt{t}]})$; update $\boldsymbol{\eta}_{\mathtt{k}}$.

Functions $\mathtt{U}_{\mathtt{jk}}(\cdot)$ could either return proposals from a MCMC sampler or updates from an optimizing algorithm.

R package bamlss

The package is available at

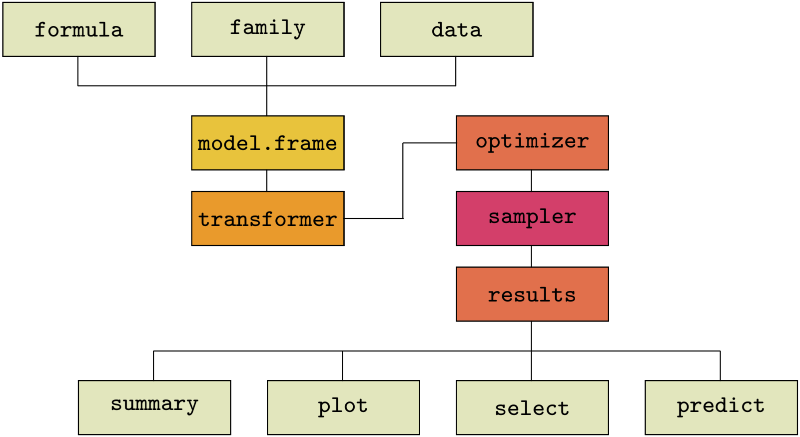
`https://R-Forge.R-project.org/projects/BayesR/`

In R, simply type

```
R> install.packages("bamlss",
+    repos = "http://R-Forge.R-project.org")
```

R package bamlss

Building blocks



In principle, the setup does not restrict to any specific type of engine (Bayesian or frequentist).

R package bamlss
Available building blocks

Type	Function
Parser	`bamlss.frame()`
Transformer	`bamlss.engine.setup()`, `randomize()`
Optimizer	`bfit()`, `opt()`, `cox.mode()`, `jm.mode()`
Sampler	`GMCMC()`, `JAGS()`, `STAN()`, `BayesX()`, `cox.mcmc()`, `jm.mcmc()`
Results	`results.bamlss.default()`

To implement new engines, only the building block functions have to be exchanged.

R package bamlss

Available families

Work in progress . . .

Function	Distribution
`beta.bamlss()`	Beta distribution
`binomial.bamlss()`	Binomial distribution
`cnorm.bamlss()`	Censored normal distribution
`cox.bamlss()`	Continuous time Cox-model
`gaussian.bamlss()`	Gaussian distribution
`gamma.bamlss()`	Gamma distribution
`jm.bamlss()`	Continuous time joint-model
`multinomial.bamlss()`	Multinomial distribution
`mvn.bamlss()`	Multivariate normal distribution
`poisson.bamlss()`	Poisson distribution
. . .	

New families only require density, distribution, random number generator, quantile, score and hess functions.

Wrapper function:

```
bamlss(list(accel ~ s(times), sigma ~ s(times)),
  family = "gaussian", data = mcycle, optimizer = bfit,
  sampler = GMCMC)
```

Standard extractor and plotting functions:

`summary()`, `plot()`, `fitted()`, `residuals()`, `predict()`, `coef()`, `logLik()`, `DIC()`, `samples()`, ...

Example

Joint modeling of longitudinal and survival data

The hazard of an event at time *t* can be described with a relative additive risk model of the form:

$$\lambda(t) = \exp\left(\eta(t)\right) = \exp\left(\eta_\lambda(t) + \eta_\gamma\right),$$

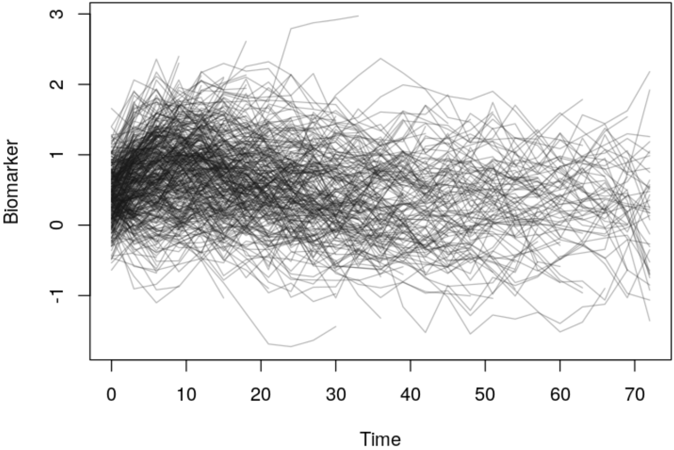
i.e., a model for the instantaneous event rate conditional on the event did not happen before time *t*.

The probability that an event will occur after time *t* is

$$S(t) = Prob(T > t) = \exp\left(-\int_0^t \lambda(u)du\right).$$

Example

In a joint-model setting, additional longitudinal data available, e.g., a biomarker measured at regular/irregular intervals.



Example

Is the risk of an event associated with the longitudinal process?

$$\lambda(t) = \exp\left(\eta(t)\right) = \exp\left(\eta_\lambda(t) + \eta_\gamma + \eta_\alpha(t) \cdot \eta_\mu(t)\right).$$

Assuming conditional independence, the corresponding log-likelihood is

$$\ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) = \ell(\boldsymbol{\beta}_{\text{surv}}; \mathbf{y}_{\text{surv}}, \mathbf{X}_{\text{surv}}) + \ell(\boldsymbol{\beta}_{\text{long}}; \mathbf{y}_{\text{long}}, \mathbf{X}_{\text{long}})$$

For estimation, compute:

- $\mathbf{s}(\boldsymbol{\beta}_\lambda)$, $\mathbf{s}(\boldsymbol{\beta}_\gamma)$, $\mathbf{s}(\boldsymbol{\beta}_\mu)$, $\mathbf{s}(\boldsymbol{\beta}_\sigma)$ and $\mathbf{s}(\boldsymbol{\beta}_\alpha)$.
- $\mathbf{H}(\boldsymbol{\beta}_\lambda)$, $\mathbf{H}(\boldsymbol{\beta}_\gamma)$, $\mathbf{H}(\boldsymbol{\beta}_\mu)$, $\mathbf{H}(\boldsymbol{\beta}_\sigma)$ and $\mathbf{H}(\boldsymbol{\beta}_\alpha)$.

I.e., posterior mode estimation with NR and derivative based MCMC.

Note, the log-likelihood and derivatives include integrals, which need to be computed numerically.

Example

In R, posterior mode estimates with function `jm.mode()`, MCMC sampling with function `jm.mcmc()`.

Both functions use the infrastructures provided by `bamlss.frame()`.

Formula for the JM

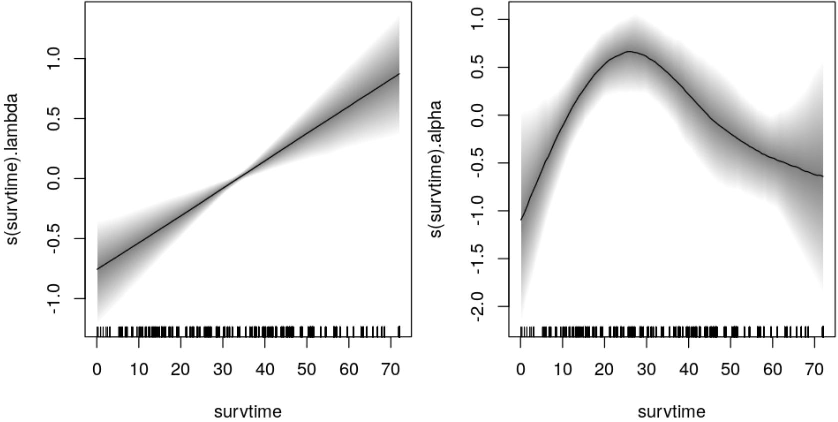
```
R> f <- list(
+   Surv2(survtime, event, obs = y) ~ s(survtime),
+   gamma ~ s(x1),
+   mu ~ s(x2) + ti(obstime) + ti(id,bs="re") +
+     ti(id,obstime,bs=c("re","cr"),k=c(nlevels(d$data$id), 5)),
+   sigma ~ 1,
+   alpha ~ s(survtime)
+ )
```

The model is estimated with

```
R> b <- bamlss(f, data = d, family = "jm",
+   timevar = "obstime", idvar = "id",
+   n.iter = 12000, burnin = 2000, thin = 40, cores = 4)
```

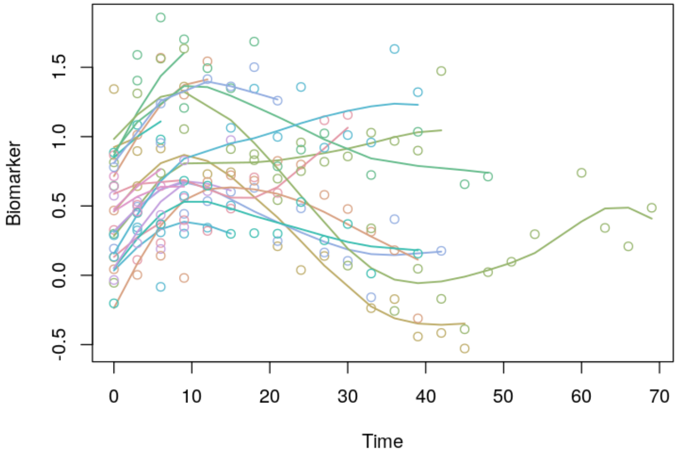
Example

```
R> plot(b, model = c("lambda", "alpha"))
```



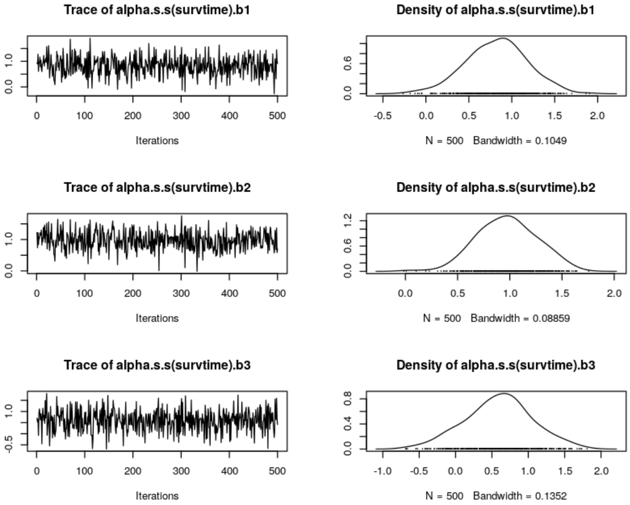
Example

```
R> nd <- subset(d, id %in% c(1:20))
R> nd$fit <- predict(b, newdata = nd, model = "mu",
+   term = c("s(x2)", "ti(obstime)", "ti(id)", "ti(id,obstime)"))
```



Example

```
R> plot(b, model = "alpha", which = "samples")
```



Thank you!!!

Klein N, Kneib T, Klasen S, Lang S (2014). *Bayesian Structured Additive Distributional Regression for Multivariate Responses.* Journal of the Royal Statistical Society: Series C (Applied Statistics), pp. URL `http://dx.doi.org/10.1111/rssc.12090`

Lang S, Umlauf N, Wechselberger P, Harttgen K, Kneib T (2013): Multilevel structured additive regression. *Statistics and Computing*, 24(2), 223–238.

Umlauf N, Adler D, Kneib T, Lang S, Zeileis A (2015). *Structured additive regression models: An R interface to* ***BayesX***. Journal of Statistical Software, 63(21):1–46, 2015.
URL `http://CRAN.R-project.org/package=R2BayesX`

Rigby RA, Stasinopoulos DM (2005). *Generalized Additive Models for Location, Scale and Shape (with Discussion).* Applied Statistics 54, 507–554.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, Boca Raton.

Wood SN (2011). ***mgcv****: GAMs with GCV/AIC/REML Smoothness Estimation and GAMMs by PQL*. R package version 1.7-6. URL `http://CRAN.R-project.org/package=mgcv`